LoadModule

Use fully-qualified path/filename to ensure that LoadModule() will load the correct file (LoadModule() is deprecated)

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-26

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 4441 bytes

<table>
<tr><td><b>Attack Category</b></td><td><ul><li>Path spoofing or confusion problem</li></ul></td></tr>
<tr><td><b>Vulnerability Category</b></td><td><ul><li>Indeterminate File/Path</li><li>Process management</li></ul></td></tr>
<tr><td><b>Software Context</b></td><td><ul><li>Process Management</li></ul></td></tr>
<tr><td><b>Location</b></td><td><ul><li>winbase.h</li></ul></td></tr>
<tr><td><b>Description</b></td><td><p>If the deprecated LoadModule() function must be used, precautions should be applied to ensure that the correct file is found.</p><p>The LoadModule function loads and executes an application or creates a new instance of an existing application. If a full path is not specified, heuristics are used to search for the application. Relying on these search rules increases vulnerability to being spoofed by a malicious executable.</p><p>This is a deprecated 16-bit API. This function is provided only for compatibility with 16-bit versions of Windows. Applications should use the CreateProcess function.</p></td></tr>
<tr><td><b>APIs</b></td><td><table><tr><th>Function Name</th><th>Comments</th></tr><tr><td>LoadModule</td><td></td></tr></table></td></tr>
<tr><td><b>Method of Attack</b></td><td>An attacker could inject a Trojan horse executable into the system by placing a "tainted" executable in a location in the search path that is found before the intended executable.</td></tr>
<tr><td><b>Exception Criteria</b></td><td></td></tr>
<tr><td><b>Solutions</b></td><td><table><tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr><tr><td>When an application needs to be loaded.</td><td>Since this function simply calls CreateProcess, the following</td><td></td></tr></table></td></tr>
</table>

[1] http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

		security guidelines apply to LoadModule: If lpModuleName is not NULL, it should be fully qualified along with file extension (i.e. exe, com, bat, cmd). Otherwise, the current directory and .exe are assumed as location and extension. Consider not passing NULL for lpModuleName to avoid the function's having to parse and determine the executable pathname from lpParameterBlock->lpCmdLine. Otherwise, use quotations around the executable path in lpParameterBlock->lpCmdLine, if lpModuleName is NULL Don't specify NULL for the lpParameterBlock->lpEnvAddress, as the new process inherits the environment of the calling process. Only the minimum set of required	

<table>
<tr><td></td><td><table><tr><td></td><td>environment variables should be passed to the child process.</td><td></td></tr></table></td></tr>
<tr><td><b>Signature Details</b></td><td>DWORD LoadModule(
LPCSTR lpModuleName,
LPVOID lpParameterBlock
);</td></tr>
<tr><td><b>Examples of Incorrect Code</b></td><td>LOADPARMS32 params;
params.lpEnvAddress = NULL;
params.lpCmdLine = "";
params.lpCmdShow = (2 << 16) | SW_SHOWDEFAULT;
params.dwReserved = 0;
DWORD result = LoadModule(TEXT("MyModule"), params);</td></tr>
<tr><td><b>Examples of Corrected Code</b></td><td>/* If one must use LoadModule() the following is safer; but using CreateProcess() would be preferred */
LOADPARMS32 params;
params.lpEnvAddress = "\0"; // terminated environment variable block with no entries
params.lpCmdLine = "";
params.lpCmdShow = (2 << 16) | SW_SHOWDEFAULT;
params.dwReserved = 0;
DWORD result = LoadModule(TEXT("C:\\MyDir\\MyModule.exe"), params);</td></tr>
<tr><td><b>Source Reference</b></td><td>• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dllproc/base/loadmodule.asp[2]</td></tr>
<tr><td><b>Recommended Resource</b></td><td></td></tr>
<tr><td><b>Discriminant Set</b></td><td><table><tr><td><b>Operating System</b></td><td>• Windows</td></tr><tr><td><b>Languages</b></td><td>• C
• C++</td></tr></table></td></tr>
</table>

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

[1] mailto:copyright@cigital.com